by Andrew Mitz & Jon Daley

DOS in the 21st Century

A USB Flash Drive Reader for MCUs Works for DOS

With a little effort, you can turn DOS into a handy real-time operating system. As Andrew and Jon explain, it can be the perfect fit for embedded applications that may require too many resources for a single-board microcontroller. Read on to learn how this USB flash drive reader for DOS can enhance your future embedded applications.

Even the smallest embedded project can read and write USB flash memory drives thanks to the Vinculum VDRIVE2 flash memory drive reader module from Future Technology Devices International. The VDRIVE2 module, with its built-in USB socket, snaps into the front panel of your project and talks to a microprocessor through either a SPI port or a serial port (see Photo 1). All of the complexities of talking to a flash memory drive (attachment, FAT16 support, and so on) are handled by its FTDI VNC1L flash memory drive host controller chip, which provides simple commands for directory listings, file transfers, and other operations. You can get a VDRIVE2 off-the-shelf from Mouser Electronics for about $25. Although the VDRIVE2 was intended for microcontroller use, our embedded application uses DOS. Much of this article will apply to both.

WHY DOS?

DOS may be dead in the desktop computing world, but it lives on as an important operating system for embedded applications. One of the great advantages of DOS is that small tweaks can turn it into a true real-time operating system that is perfect for low-volume, cost-sensitive applications that require too many resources for a microcontroller. At the National Institute of Mental Health (part of the NIH), Andrew uses an open-source data acquisition and control program called NIMH Cortex. It is a real-time DOS application developed for behavioral brain research. It supports low-speed analog channels (each with a 1 ksps conversion rate), lots of digital I/O, and a few specialized interfaces (e.g., touchscreens), and it comes with a companion program for near

Photo 1—The Vinculum VDRIVE2 USB flash memory drive reader snaps easily into a front panel. It has a built-in microprocessor that manages the flash memory file system with simple serial or SPI port commands, so just about any embedded controller can access a flash memory drive.

real-time display of simple graphics objects on a separate computer. You program NIMH Cortex in C. Once you get past the learning curve, it is a powerful tool for automating experiments. One thing we really miss on our DOS system, however, is a USB flash memory drive reader. USB disk support can run under DOS (refer to www.bootdisk.com/usb.htm for examples), but large applications like NIMH Cortex don't tolerate the terminate and stay resident (TSR) and other drivers that eat up scarce DOS resources. We have circumvented the DOS memory problem by using the VDRIVE2. In this article, we will describe our mixed hardware/software solution for a DOS flash memory drive reader/writer that can be assembled for less than $50 and doesn't use resident memory.

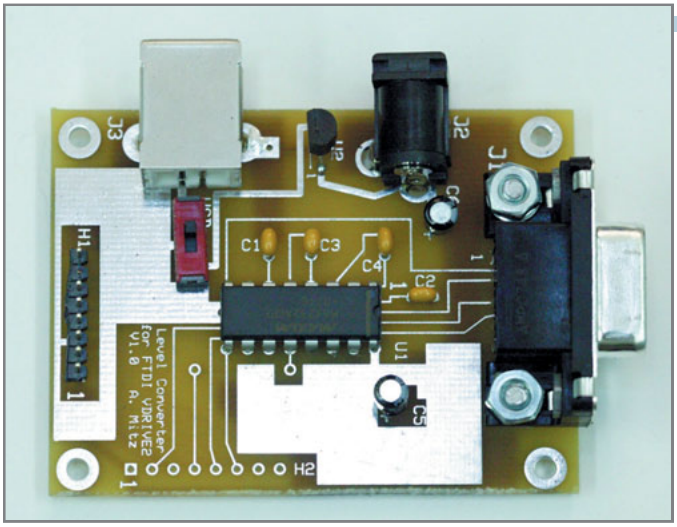


Photo 2—The VDRIVE2 has 5-V (TTL/CMOS) signals. It needs a level converter to interface directly with a serial port. Our level converter circuit board is shown here. It has a header for the VDRIVE2 on one end and a board-mounted serial port connector (DE-9) on the other end.

HARDWARE

The VNC1L chip has a built-in UART, but the chip provides only 5-V logic signals. RS-232 serial ports require ±5 to ±15 V, so a level converter is needed for the VDRIVE2. A Maxim Integrated Products MAX232A is the most common chip for this job. It needs only four 0.1-µF capacitors and a decoupling capacitor to provide buffering and bipolar voltage boost. We designed a small PCB using the free layout tools from ExpressPCB. Two copies of the circuit board layout fit onto a single 2.5″ × 3.8″ board, the size used for the ExpressPCB MiniBoard Service. Using our layout file, you can electronically order three boards (six copies of the circuit) for about $60. The layout file is posted on the *Circuit Cellar* FTP site. Alternatively, you can delete one copy of our circuit from the layout and lay out another project. Then your $60 will get you three converter boards and three of your design. Just don't forget to leave space between the layouts for cutting each board. If you plan to use the VDRIVE2 with a SPI port, a level converter is not necessary. However, a level converter will enable you to use your computer as a test environment for learning about the VDRIVE2 and even for embedded code development.

Photo 2 shows the populated circuit board. The schematic is in Figure 1. Table 1 has a complete description of the parts, including the parts for power and packaging. A nine-pin D-sub connector (J1, which is technically a DE-9, but often called a DB-9) mounts directly on one end of the PCB for connection to the computer serial port. The other end of the circuit board has a single-row 2-mm pitch header (H1). The header matches the jumper cable that comes with the VDRIVE2. For tight spaces, you can cut one connector off of the jumper cable and solder the cut wires directly to the circuit board. An alternative header socket (H2) is on the side of the circuit board; it has the same connections as H1, but with 0.1″ spaced pins. H2 can be used for soldering the cut VDRIVE2 wires or to gain access to the VDRIVE2 signals for testing. Like the two connector alternatives, two alternatives are provided for powering the board, as well. The layout has holes for a circular connector (J2) commonly used for wall power transformers (bricks). Any DC brick rated for 9 to 15 V at 100 mA will work. A low-power, three-terminal regulator (LM78L05, U2) drops this voltage down to 5 V for the MAX232A and the VDRIVE2. Alternatively, regulated 5 V can be pilfered from any available USB port via J3. J3 is a type-B USB connector,

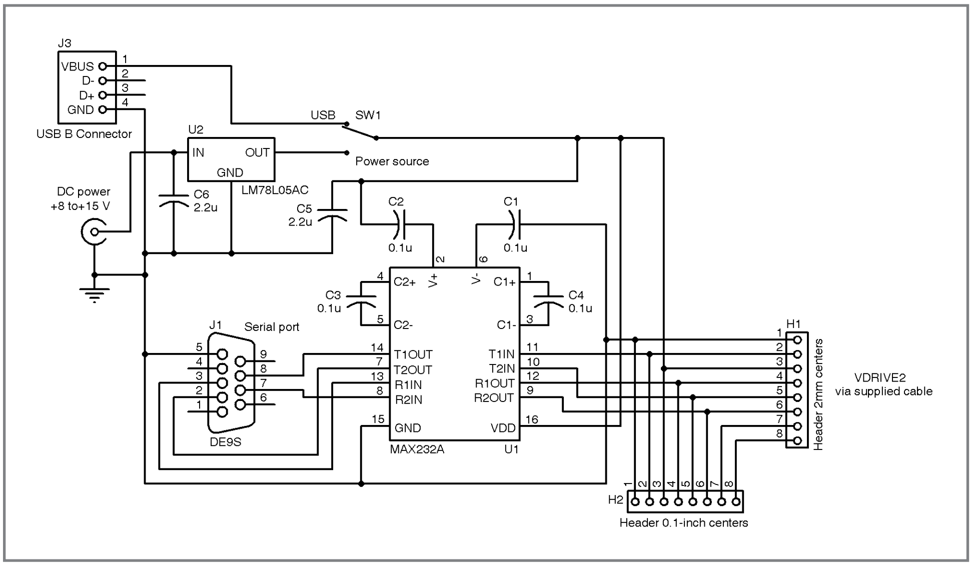


Figure 1—The level converter schematic shows two power options: wall transformer or USB port power. The circuit can be assembled for one or both power options.

Quantity	Ref	Description	Source	Part Number	Notes
1		Vinculum VDRIVE2 USB flash memory drive interface	Mouser Electronics	895-VDRIVE2	
2	C1, C2, C3, C4	0.1-µF/50-V Ceramic capacitor	Digi-Key	399-4264-ND	
3	C5, C6	2.2-µF/50-V Miniature electrolytic capacitor	Digi-Key	P825-ND	
1	H1	Single-row 2-mm header, 50 pins	Digi-Key	SAM1176-50-ND	Break off eight pins for board
3	H2	Single-row 0.1″ header			Do not install a connector
2	J1	Nine-position female D connector, PC mount, right angle	Digi-Key	A32075-ND	
1	J2	DC power connector, PC mount, 2.1 × 5.5 mm	Digi-Key	CP-202A-ND	Mount connector only if using a wall transformer
1	J3	USB B connector, PC mount	Digi-Key	AE9925-ND	Do not mount connector if only using a wall transformer
1	SW1	Ultra-miniature slide switch	Digi-Key	360-2133	Use jumper instead of switch if only one of J2 or J3 is used
1	U1	MAX232A RS232 Driver/receiver, 16-pin DIP	Digi-Key	MAX232ACPE+-ND	
1	U2	LM78L05AC Low-power 5-V linear regulator, TO-92	Digi-Key	MC78L05ACPFS	
2		Mounting screws for J1, 4-40 × 3/8	Digi-Key	H781-ND	Box of 100, only two needed
2		Nut 4-40	Digi-Key	H216-ND	Box of 100, only two needed
1		D-Sub hardware set	Digi-Key	609-1420-ND	
1		Printed circuit board	ExpressPCB		Order using ExpressPCB software and .PCB file
1		Wall transformer 9 VDC at 125 mA	Digi-Key	MT7141-ND	
1		Serpac series A plastic case, black	Digi-Key	SRA21B-ND	
1		End panel with DB9 cutout for Serpac A-21	Digi-Key	SR2005-DB9B-ND	

Table 1—This is a complete listing of parts and parts sources. A number of parts (J2, C6, U2, SW1, wall transformer) can be omitted when using the computer's USB source for power.

making it easy to bring power through a standard USB peripheral device cable. When using the USB port for power, J2, U2, and C6 are not needed. Switch SW1 is shown for those who might want both options on the same board, but more commonly a board will be assembled for only one power option with a wire jumper in place of S1.

You might want to mount the VDRIVE2 separately from the level converter board, but we chose to package them together in a plastic box (Serpac Series A) (see Photo 3 and Photo 4). A wide rectangular slot is cut into one end plate of the box to accommodate the VDRIVE2. The other end plate is replaced with one precut for the DE-9 (Serpac A-21) (see Table 1). Standard DE-9 hardware secures the connector to the end plate and provides a threaded fastener for the connecting cable. The arrangement is a tight fit and some of the internal plastic ribs and stand-offs must be trimmed with a hobby knife, but the final product is appealing. The end plates and remaining stand-offs provide plenty of support for the circuit board and VDRIVE2. No additional mounting screws are needed. Cuts in the box's sloping sidewalls provide access to the power connectors.

Look at the VDRIVE2 datasheet before completing your assembly. The three-pin jumper on the back of the device (UART/SPI) should be set for pull-up. Once assembled, connect one of the power sources and connect your computer's serial port. You can use a modem program in either DOS or Windows (e.g., HyperTerminal) running at 9,600 bps to chat with the VDRIVE2. You can even plug in a flash memory drive and get a directory listing with simple text commands. Refer to the Firmware manual for examples.

SOFTWARE CONSIDERATIONS

Our reason for using special hardware was to preserve DOS memory by avoiding resident drivers. Thus, from the outset, the plan was to produce a set of DOS commands for each aspect of talking to the flash memory drive (read, write, directory listing, and more). These primitive commands could be used on their own or serve as the backbone for an alternative user interface, perhaps a Norton Commander-style interface.

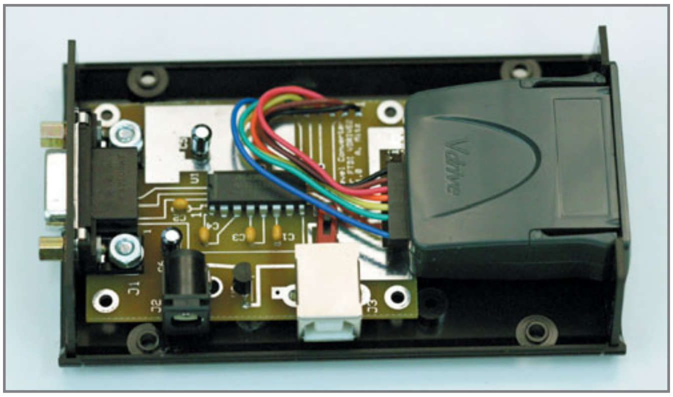


Photo 3—This shows how we packaged the VDRIVE2 together with the level converter circuit board. The fit is tight, so the VDRIVE2 cable is wired directly into the H2 pads rather than through a connector.

Saelig

UNIQUE PRODUCTS & SUPPORT

www.saelig.com

Testgear

Color LCD Scope

2-ch + trigger standalone USB bench scope. **$325 / $599**

PDS5022S / PDS6062T

2-ch 1GSa/s Scope

2-ch 1GSa/s (25GSa/s equiv.) 50/100 MHz scope. **$595 / $795**

DS1000E

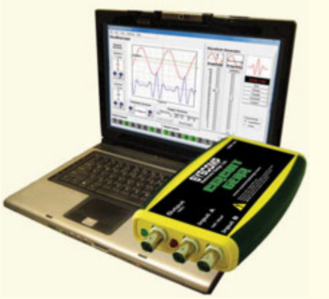
Scope + Analyzer

25MHz 2-ch /16 logic scope and logic analyzer. **$699**

DS1022CD

Amazing 7 in 1 Scope! $180

CircuitGear CGR-101™ is a unique new, low-cost PC-based instrument which provides the features of seven devices in one USB-powered compact box: 2-ch 10-bit 20MSa/sec 2MHz oscilloscope, 2-ch spectrum-analyzer, 3 MHz 8-bit arbitrary-waveform/standard-function generator with 8 digital I/O lines. It also functions as a Network Analyzer, a Noise Generator and a PWM Output source – all for less than $180! What's more – its' open-source software runs with Windows, Linux and Mac OS's! **Only $180**



CGR-101

Handheld Scope

20MHz / 60MHz rugged handheld USB 2-ch scope. **$593 / $699**

HDS1022MN / HDS2062M

Pen Scope

10/25MHz USB powered scope-in-a-probe! Up to 100MS/s. **$193 / $308**

PS2104/PS2105

6 in 1 Scope

200kHz 2-ch 10-bit scope, 2-ch spectrum analyzer, 16-ch 8MHz logic analyzer, 5-ch sig gen, 8-ch pattern gen. **$199**

PoScope with Probes

Low-Cost Scope

2-ch 40/100/200MS/s 8-bit scope range with 5/10/25MHz. **$297 +**

PS2203/4/5

1/2GHz RF Generators

High accuracy/stability, wide range, low phase noise/leakage, serial control.

TGR1040 / TGR2050

Mixed-Signal Scope

100MHz Scope, + Spectrum/Logic Analyzer and Signal Generator. **$1259+**

CS328

USB Bus Analyzers

Packet-Master™ - USB 1.1/2.0 analyzers and generators. **$699 +**

USB12 / 480+ / 500AG

16-Ch Logic Analyzer

Intuitive full-featured 16-ch 4MB 200MHz sampling memory. **$299**

LAP-16128U

SPI Bus Analyzer

Protocol exerciser/analyzer for standard SPI and non-standard 4-wire and 3-wire serial protocol interfaces up to 50 Mbps.

SPI Xpress

EMC Spectrum Analyzer

RF & EMF Spectrum Analyzer 1Hz to 7GHz for WiFi, mikes, etc.

EMC RF & EMC Spectrum

EMC Spectrum Analyzer

Handheld Palm PC-based 2.7GHz Spectrum Analyzer.

PSA2701T

"I really like this scope adapter - it's really meant for teaching electronic experiments but it's ideal for engineers too."

Alan Lowne
Saelig CEO

www.saelig.com

Waveform Generator

USB2.0 speed 16-bit digital pattern or arbitrary waveform generator.

Wave Xpress

I2C Xpress

Versatile USB 2.0 I2C protocol exerciser and analyzer.

I2C Xpress

Automotive Testing

Kits turn your PC into vehicle-electrics diagnostic tool.

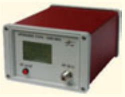
PS3423 / KLARI-MOD

CAN Gateway

Janz - Full-featured standalone fanless industrial Linux PC.

emPC-x133

RF Generator



High-res, extremely low-noise, portable 3GHz RF generator.

APSIN3000

RF Testing/EMI Tents

Portable RF test enclosures & shielding tents with external frame.

RF Testing / EMI Tents

Wireless Data Loggers

Log and display temp, hum, volt, event-time or pulse-counting data

RTR-50

Multiparameter Loggers

Mini-logger with built-in temp/hum/pressure/3-axis accel sensors.

MSR145S

USB Logger

Standalone USB temp / hum / volt / current loop data logger. **$49+**

EL-USB-1/2/3/4

Electronic DC Load

Const. current, resistance, conductance, voltage & power modes

LD300

60/100/120MHz AWG

60/100/120MHz USB 14-bit ARB with USB RS-232, LAN/GPIB.

DG3061A/3101A/3121A

TorqSense

Configurable, patented USB-output non-contact SAW digital rotary torque transducers with integral electronics.

RWT320

Misc

USB to I2C

"Drop-in" solution connects PC to I^{2}C/SMBUS + 32 I/O lines. **$89**

USBI2CIO

FTDI USB ICs

Popular UART and FIFO chips. Upgrade Legacy designs to USB.

FT232RL

CAN-USB

Intelligent CAN connection from PC's USB port. **$299**

CAN-USB

Serial-Ethernet Cable

Network serial product easily without a PC using this 28" cable. **$89**

eCOV-110-P

Lorlin Switches

Fantastic array of stock and custom switching devices.

Lorlin

PSoC Starter

Get going quickly with PSoC visual design environment.

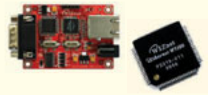
PSoC Starter

Keyboard Simulator

USB board adds 55 I/O and 5 x 10-bit A/D inputs, 1 x 10-bit analog O/P.

PoKeys55T

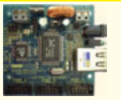
Instant Ethernet



No OS needed. TCP/IP offload, ICs improve system performance.

WIZ110SR / W5100

Ethernet-IO



UDP/IP-controlled 24 digital I/O board 3 x 8-bit TTL ports.

Ether-IO 24

FPGA Systems

Ready-to-go out-of-the-box FPGA/DSP designs for beginners and experts!

RTG005

Wireless Solutions

Analog input, bluetooth wireless modules 433/868/915MHz.

EmbedRF / Adeunis

Temp/RH Sensors

Novel ambient sensors & modules accurately measure temp/RH.

UPSICAP / DLP-TH1

.NET Board

Small (2.2" x 2.2") lowest cost .NET Micro Framework dev system.

USBizi

Easy OLED Display

Compact, economical smart OLED with graphics drive from USB or RS232.

uOLED-96-G1

RF Modules

Simultaneously transmit composite video and stereo audio signals.

AWM6XX TX/RX

RS232 to 422/485

9p-9p or 25p-25p self-pwrd, isolated RS232-RS422/485

KK Systems

1/2/4/8/16 x RS232

Add 1-16 COMports via your PC's USB Port easily.

USB-COM

Quantum

Quickly add capacitive touch on / off & X / Y - sensing ICs.

Quantum

Saelig .com
unique electronics
888-75AELIG info@saelig.com

*Above are some of our best-selling, unique, time-saving products - see our website for 100s more: WiFi/910MHz antennas, wireless boards, LCD display kits, Ethernet/IO, USB/RS232/485, USB-OTG, instant Ethernet-serial, CAN/LINbus, USB cables/extenders, line testers, logic analyzers, color sensors, motion controllers, eng. software, wireless boards, SMD adapters, I2C adapters, GPS loggers, automotive testing, security dongles, video motion detectors, crystals/oscillators, custom switches, barcode scanners, DSP filters, PLCs, Remote MP3 players, etc. FREE Starbucks card with your $50 order! Check **www.saelig.com** often for special offers, bargains, business hints, blog, etc.*

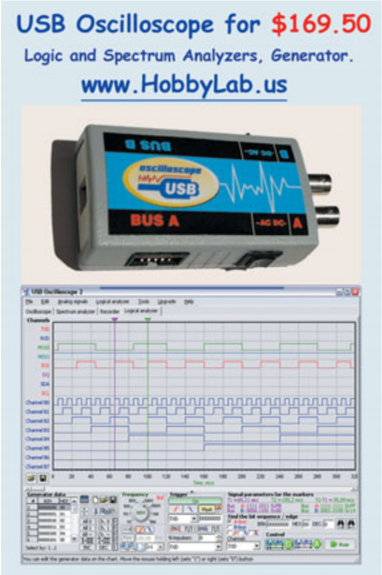


Photo 4—This is the complete assembly, with the VDRIVE2 on one end and the level converter power connectors accessible from the side. The serial port connection (not visible) is opposite the VDRIVE2.

A number of issues arose before and during development that were related to the problem of operating without a resident driver. A resident driver can remember the data rate and other serial port settings entered earlier by the user. Programs not loaded into memory need another way to avoid Korsakoff's syndrome. The two standard options are environmental variables and configuration files. We chose to use environmental variables. You can place SETENV commands in the AUTOEXEC.BAT file for boot-up configuration, but we recommend using our batch file that takes the parameters on the command line and checks for valid environmental variable names.

Environmental variables provide memory for the different programs using the VDRIVE2, but things get messy when trying to change the data rate of the VDRIVE2. If the VDRIVE2 has been set to one baud rate (e.g., 9600) and then you change the environmental variable to a new baud rate, the old baud rate information is lost. Multiple environmental variable sets could be employed—one for program tracking and one for user requests, but this approach was eschewed out of personal preference. Rather, we chose an algorithm that hunts for the proper data rate if communication with the VDRIVE2 is lost. The algorithm is useful for other occasions, such as when the VDRIVE2 is first powered up.

Perhaps the most vexing problem associated with implementing read-write programs for the VDRIVE2 is dealing with the directory structure of the flash memory drive. The VDRIVE2 does not keep track of the current default directory. Without a resident program, it is difficult, albeit not impossible, to keep track of the current directory. Our programs require you to take responsibility for knowing the current default directory after one or more change directory commands. To complicate matters, the DRIVE2 supports only 8.3 (eight-character alphanumeric name followed by up to three-character alphanumeric extension) DOS file names. Thus, Windows XP directory names will often look like PROJEC~2. In our application, these limitations do not pose any real inconvenience. We use simple directory structures to move files from our NIMH Cortex data acquisition system to a Windows computer for editing or data analysis. With this in mind, we did not implement complex directory parsing at the command line. Specifications like ..\..\PJM\DAY2 are not supported, although specifications like ..\DAY2 and \P\PJM\DAY2 are.

The VDRIVE2 firmware must be at version V3.64 or later. Updating the VDRIVE2 can be done in two ways. One, you can download VPROG reflasher COM utility and the latest VDAP ROM file from the Vinculum download page on the FTDI web site. The Reflasher COM utility runs from DOS or a DOS window and programs the VDRIVE2 via the serial port. Alternatively, you can rename the FTD file for the latest release to FTRFB.FTD, put the file on the root directory of a flash memory drive, and just plug the flash memory drive into the VDRIVE2. The VDRIVE2 will find the file and update its own firmware.

SOFTWARE IMPLEMENTATION

The eight DOS commands for using the VDRIVE2 are in Table 2. Two commands, FCONFIG and FBAUD, provide ways to establish the serial port communications parameters with the VDRIVE2. FCONFIG is the only command implemented as a DOS batch file (.BAT). The other commands are executable files (.EXE). FCONFIG sets environmental variables in a convenient way, providing some helpful error checking. Here are two example FCONFIG commands:

```
fconfig com 2 baud 115200
fconfig address 0x2E8 irq 5 baud 9600
```

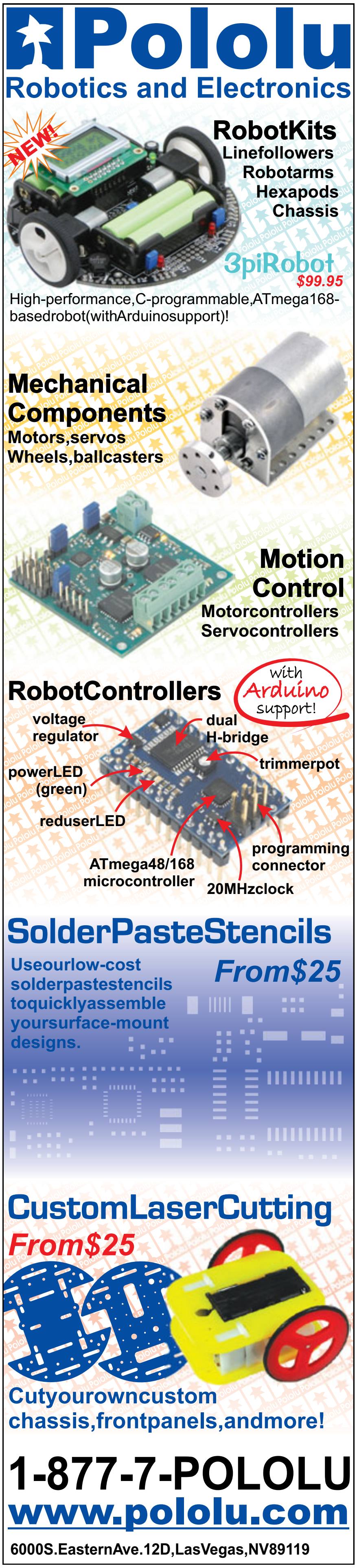
The first example selects COM port 2 at 115,200 bps. The port and interrupt addresses are the standard ones for COM port 2. The second example explicitly sets (nonstandard) addresses. FCONFIG does not try to initiate communications with the VDRIVE2, FBAUD does. FBAUD initiates a search for the VDRIVE2 device, tries to establish a new data rate, then reestablishes communications at the new rate. Allowed data rates are 2,400, 9,600, 19,200, 38,400, 57,600, and 115,200. A power-up reset places the VDRIVE2 at 9,600 bps.

FCD is used to change the default directory on the flash memory drive. When first inserted, the flash memory drive is set to the root directory. FCD can return the flash memory drive to its root directory using just a backslash as the command line parameter, just like the DOS CD command. Other operations are similar to the DOS CD command, with two exceptions, as noted above. FCD does not try to manage complex tree commands, and FCD will not report the current default directory path. FDIR also does not report the current directory path. FDIR with no command line parameters lists the entire contents of the current directory without any further details. FDIR with a specific file name will display the file size of that file. Simple DOS wild cards will work as expected (e.g., `FDIR *.DAT`), but more complex wild card constructs are not implemented. This limited directory functionality could be greatly expanded. We encourage you to think about writing a more convenient user interface using our source code as a starting point.

The file operation commands are FPUT, FGET, FDEL, and FREN. We thought about implementing an FCOPY command rather than FPUT and FGET, but having separate commands lets us use the file path parsing capabilities of DOS. Access to files on the DOS system can use any legal

DOS Command	Purpose
FCONFIG.BAT	Set COM port address, data rate, and IRQ in the environment.
FBAUD.EXE	Establish communications with VDRIVE2 at selected data rate.
FCD.EXE	Change default directory.
FDIR.EXE	List current default directory.
FPUT.EXE	Copy a file from the current DOS directory to the flash memory drive.
FGET.EXE	Copy a file from the flash memory drive to the current DOS directory.
FDEL.EXE	Delete a file on the flash memory drive.
FREN.EXE	Rename a file on the flash memory drive.

Table 2—This table lists the eight DOS commands for accessing files on the flash memory drive. FCONFIG is a batch file; the others are all exactly the same .EXE files with different names!



> A combined hardware/software solution provides an affordable way to read from and write to flash memory drives while in DOS.

file path construction, while access to files on the flash memory drive are limited to the current default directory of the flash memory drive. Once again, there is an opportunity for you to program a more general-purpose interface that supports more complete file path constructions for the flash memory drive and renaming files as they are copied. FREN lets you manually change a file name on the flash memory drive and FDEL can delete a file. FPUT, FGET, and FREN will overwrite only a destination file if a /y option is added as the first command parameter. (Note that the option /h will list instructions for any of the commands.)

The most challenging part of the source code development was managing the FIFO of the UART during serial port transfers. The limited computational power of the VDRIVE2 and the wide variation in personal computer hardware require careful coding of the serial port handshaking. The FIFO of the computer UART must be enabled and disabled at critical moments to maximize throughput without overrunning the VDRIVE2 buffer at high data rates. The VDRIVE2 provides an acknowledge handshake, so errors are always trapped. Even with the careful handshake, a file copy (FPUT or FGET) will fail on occasion. These failures are rare unless the computer hardware just cannot handle one of the high data rates. For large files (over a few megabytes), it probably makes more sense to reboot the system to Windows and use the full speed of a USB port. It takes about 50 minutes to copy a 30-MB file at 115,200 bps, which is close to the theoretical minimum time (about 46 minutes).

COMPILER & SOURCE CODE

The source code is posted on the *Circuit Cellar* FTP site. Although the source code is rather generic C and should work with any C compiler, the code comes with a project file for compilation using Borland C++ version 1.1. The Borland compiler is available as a free download for noncommercial use (http://dn.codegear.com/article/21751). On the web site, you will find a lot of information for installing and using the compiler. For this project, installation on a Windows XP computer is straightforward. We use all installation defaults except the source disk (C drive instead of a floppy disk). To keep things simple, the source code is unzipped to a new directory C:\TC\FSOURCE, a subdirectory of the Turbo C++ compiler root. After everything is set up, double click the TC.EXE file in the C:\TC\BIN subdirectory to get character-based Turbo C++ GUI. If you want, use the properties option of the DOS window to switch to full-screen mode and flash back to the good old days of DOS.

Compile is as easy as install. Once Turbo C++ is running, type ALT-P to drop down the Project menu and select Open Project. Navigate to the FSOURCE directory and select MASTER.PRJ. You should end up back in the main Turbo C++ screen. Type ALT-C for the Compile menu and select Build All. That will create MASTER.EXE, the only file you need. Why is MASTER.EXE the only compiled file? Because all of the other .EXE files are just clones of MASTER.EXE with different names. When the program is started, it looks at the first parameter of the command line, which is the file name of the program; the program's file name determines what action to take. You can manually make copies of MASTER.EXE, renaming each copy to match the commands, or you can use the batch file MAKE.BAT to generate the entire set. Consolidating the object code to

Listing 1—The file name for each command is hard-coded in MAIN.C. Change the string in the `strncmp` function and recompile MAIN to rename a command.

```
if(!strncmp(szCommand, “FDIR.EXE”, 8))                   g_eCmd = COMMAND_FDIR;
else    if(!strncmp(szCommand, “FPUT.EXE”,               8))     g_eCmd = COMMAND_FPUT;
else    if(!strncmp(szCommand, “FGET.EXE”,               8))     g_eCmd = COMMAND_FGET;
else    if(!strncmp(szCommand, “FCD.EXE”,                7))     g_eCmd = COMMAND_FCD;
else    if(!strncmp(szCommand, “FREN.EXE”,               8))     g_eCmd = COMMAND_FREN;
else    if(!strncmp(szCommand, “FDEL.EXE”,               8))     g_eCmd = COMMAND_FDEL;
else    if(!strncmp(szCommand, “FTER.EXE”,               8))     g_eCmd = COMMAND_FTER;
else    if(!strncmp(szCommand, “FBAUD.EXE”,              9)      g_eCmd = COMMAND_FBAUD;
else    g_eCmd= COMMAND_UNKNOWN;
```

a single program simplifies development in many ways, but the down side is that you cannot rename the file operations. Thus, if you do not like the program name FPUT.EXE, you might be tempted to rename it to, for example, TOFLASH.EXE. However, the renamed program will just give you an error message. To change the name of a command, you will also have to change the name in the source code of MAIN.C, shown in Listing 1. Change the file name in quotes and then update the number of characters to match the new name. Do not exceed the 8.3 (12 character) DOS limit. After updating MAIN.C, repeat the Build All step above and then copy MASTER.EXE to the new command file name.

A BETTER INTERFACE

A combined hardware/software solution provides an affordable way to read from and write to flash memory drives while in DOS. In this implementation, inexpensive hardware obviates the need for resident drivers. While version 1 of the software can be a little awkward when the flash memory drive has complex directory structures, the source code and information presented here provide a pathway to better user interfaces. It is easy to envision a Norton Commander-like interface, or some other classic semi-graphical character-based user interface for selecting and operating on files and directories. We hope to hear from readers who take on this part of the challenge. Because USB ports have such blazing speeds compared to their serial predecessors, large files are best managed by setting up dual-boot operating systems that enable you to defenestrate DOS for Windows when more serious file transfers are necessary. For smaller files, the VDRIVE2 and the new DOS commands are an ideal solution.

Although the software was written for DOS, much of the code is reusable for microcontrollers with C compilers, or as an example for other embedded applications. If you build the level converter, you have a great set of tools for experimenting with the VDRIVE2. ■

Authors’ note: This research was supported by the Intramural Program of the National Institute of Mental Health, National Institutes of Health.

Andrew Mitz (arm@nih.gov) is a Ph.D. research scientist at the National Institutes of Health where he studies the electrical activity of the brain. He received his Bachelor’s and Master’s degrees in electrical engineering from Washington University and the University of Maryland, before entering a medical research program at Emory University. Andrew’s laboratory work involves instrumentation of physiological signals (e.g., muscle activity, heart rate, eye tracking, and microvolt recordings of brain cells). He has worked on the development of many real-time embedded systems, including robotics. In his free time, Andrew collects and repairs antique radios and supports emergency communications through amateur radio. He has also developed devices for people with disabilities. Some of Andrew’s designs are now in commercially shipping products, and many end up as publications in a wide variety of professional and hobby magazines. His favorite moments in embedded design are at the beginning and the end of each project. “The middle,” Andrew says, “is often quite maddening!”

Jon Daley is an embedded software engineer who currently can be found alternating between the two extremes of assembly language and ajax/php for his startup company Lime Daley. Wherever he is, you can find him at http://limedaley.com or circuitcellar@jon.limedaley.com.

PROJECT FILES

To download code, go to ftp://ftp.circuitcellar.com/pub/Circuit_Cellar/2009/226.

SOURCES

FTDI VNC1L Host controller chip and VDRIVE2 reader module
Future Technology Devices International | www.ftdichip.com

NIMH Cortex
NIMH Laboratory of Neuropsychology | www.cortex.salk.edu